

3D CAD für Linux

Feb. 2017

Solvespace

Hier ein kurzer Einblick: <https://www.youtube.com/watch?v=cZzn9Z2j2cQ>

Version 2.3 hab ich unter Linux Mint 17.3 erfolgreich kompilieren können.

- clone git-repo
<https://github.com/solvespace/solvespace/tree/v2.3>
- check out V2.3
- update sub modules
- Bibliotheken installieren, hierbei musste ich kleine Anpassungen zur README.md machen:
statt **libgl-dev** installierte ich **libgl1-mesa-dev** und
statt **libglu-dev** installierte ich **libglu1-mesa-dev**.
- Der Rest wie in der README.md beschrieben.

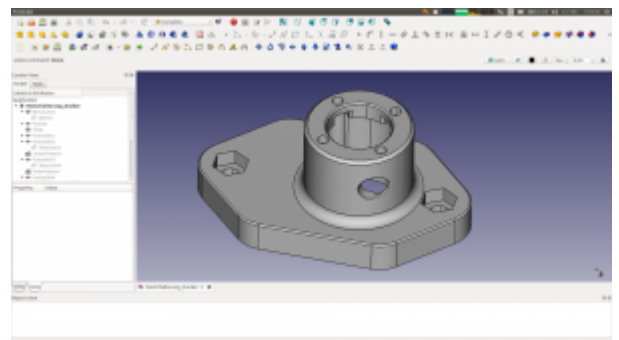
Hab bisher nur das Video-Tutorial nachgeahmt, funktioniert wirklich sehr gut.

Interessant sind auch die Beispiele hier: <http://solvespace.com/examples.pl>

Besonders fasziniert hat mich der Inversor von Peaucellier, siehe [Wikipedia](#)

FreeCAD

<http://www.freecadweb.org/>



Benutzer-Kommentar:

...

Da ist FreeCAD insgesamt schon viel weiter.

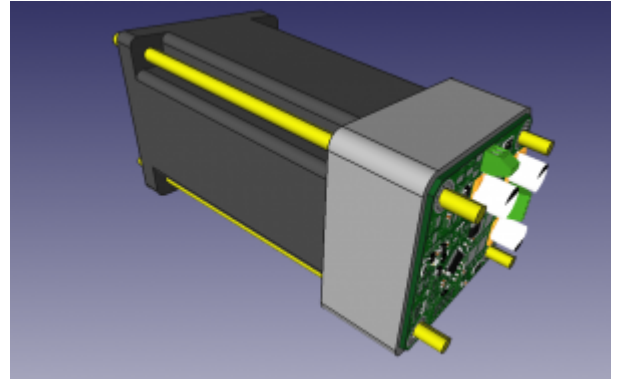
Sieh dir zb. mal das an was ich gemacht habe.

Die Baugruppe ist vollkommen parametrisch und aus mehreren einzelnen Teilen zusammengesetzt. Die wichtigsten Parameter der ganzen Baugruppe lassen sich über eine Master-Datei einstellen. Die Platine ist aus KiCad importiert.

Auch bei der Erzeugung komplexer Geometrien (zb. mit Radien und Fasen an gekrümmten Kanten) ist

FreeCAD mittlerweile ganz gut.

Besonders wertvoll ist IMHO das Modul-System von FreeCAD. Dadurch kann man recht einfach Zusatzmodule in Python oder C++ schreiben und einbinden ohne alles neu zu kompilieren. zb. Assembly2 oder Sheetmetal. Es gibt auch ein Modul für OpenSCAD. Man kann also ganze OpenSCAD Geometrien in den FreeCAD-Modellbaum integrieren. Damit wurde die „Konkurrenz“ quasi aufgefressen. Vielleicht passiert das auch mit Solvespace...



Klar, es gibt noch viele Bugs und wegen des etwas wirren Aufbaus des Abhängigkeitsbaumes bekommt man recht schnell irgendwelche Konflikte. Aber im großen und ganzen ist es für meine Zwecke schon sehr gut verwendbar.

Heeks CAD

scheint interessant zu sein und recht einfach.

Einfache Installation per PPA: <https://launchpad.net/%7Eneomilium/+archive/ubuntu/cam>

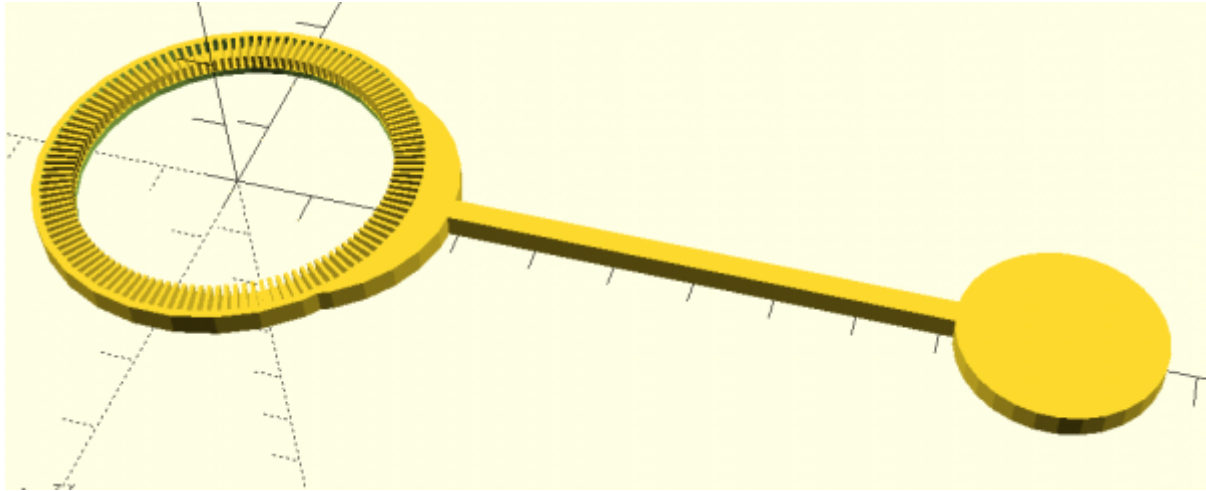
TODO

OpenSCAD

Eine skript-basierte CAD-Software.

Wenn man sich mal drüber traut, ist es dann auf einmal gar nicht so schwierig, eine Idee im Kopf in ein 3D-Modell zu verwandeln.

[Hier mein erstes Beispiel, ein Seifenblasenstiel:](#)



```

di = 40; // innendurchmesser
da = 48; // ausßendurchmesser
dr = (da-di)/2; // radius delta
speichenbreite = 0.5;
spaltMin = 0.5;

module speiche(breite, hoehe, radius) {
    translate([radius/2,0,hoehe/2]){
        cube([radius, breite, hoehe], center = true);
    }
}

// Seifenblasenseicher
module speichenring(dAussen, dInnen, z) {
    U = dInnen*PI; // Umfang innen
    n = floor(U/(speichenbreite + spaltMin));
    for(i=[0:360/n:360]){
        rotate(i, [0, 0, 1]) {
            translate([di/2,0,0]) {
                speiche(speichenbreite,z,(dAussen-dInnen)/2);
            }
        }
    }
}

module ring(dA,di,z) {
    difference(){
        cylinder(h=z,d=dA);
        cylinder(h=3*z,d=di, center=true);
    }
}

// Trägerring
translate([0,0,0]){
    ring(da, di, 1);
}

```

```
translate([0,0,1]){
    speichenring(da, di, 2);
}

module stiel(durchmesser, laenge){
    translate([laenge/2 + da/2, 0, durchmesser/2])
    cube([laenge, durchmesser, durchmesser], center = true);

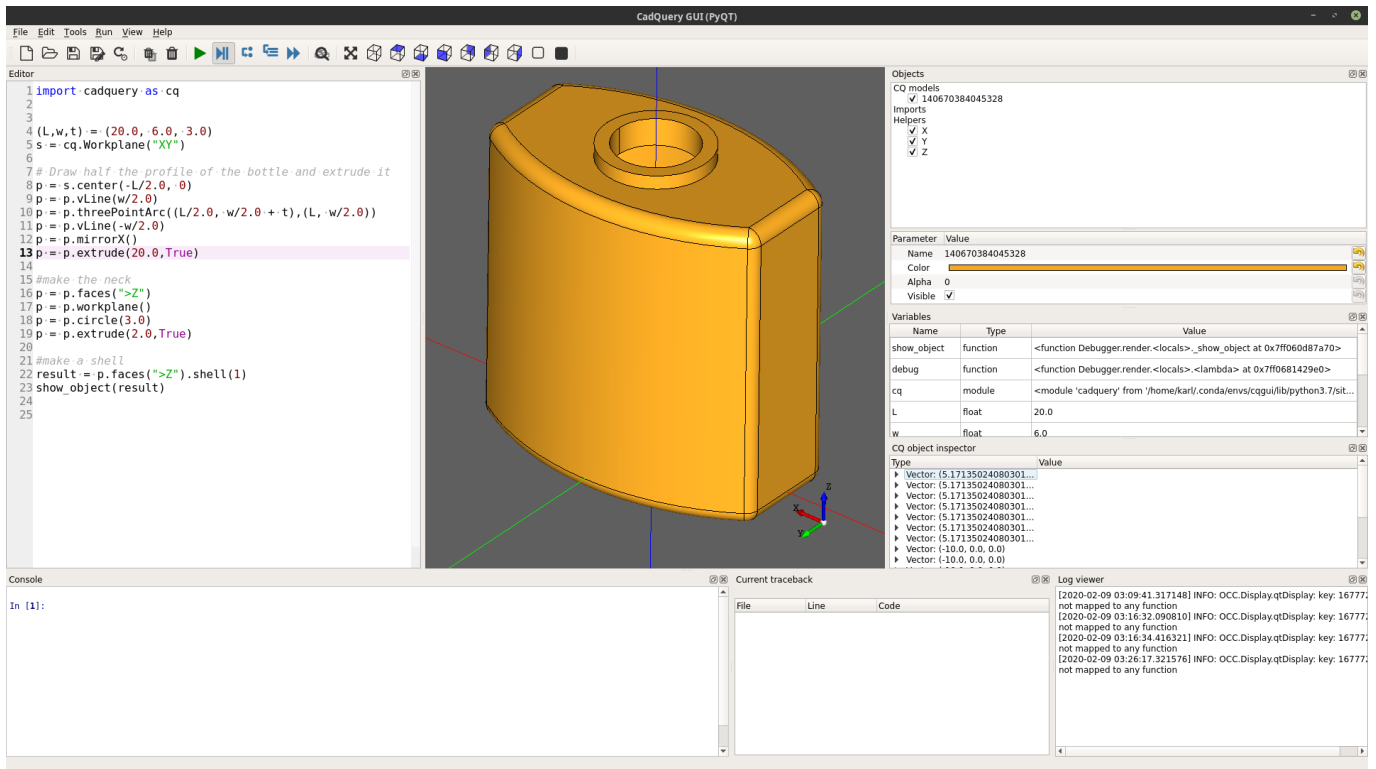
    translate([laenge + da/2, 0, durchmesser/2])
    cylinder(h=durchmesser, d=25, center = true);
}

stiel(3,80);
difference(){
    translate([10,0,0])
        cylinder(d=da-10,h=3);
    cylinder(d=da, h=10, center = true);
}
difference(){
    cylinder(d=da+2, h=3);
    cylinder(d=da, h=10, center = true);
}
```

CadQuery

Ebenso wie OpenSCAD ist CadQuery skriptbasiert. Der Vorteil hier ist, dass Open Cascade verwendet wird, und somit exakte geometrische Manipulationen ermöglicht werden. Zur Installation hab ich Anaconda in /opt/anaconda3 installiert und dann CQ-Editor installiert:

<https://github.com/CadQuery/CQ-editor/issues/116>



```

import cadquery as cq

(L,w,t) = (20.0, 6.0, 3.0)
s = cq.Workplane("XY")

# Draw half the profile of the bottle and extrude it
p = s.center(-L/2.0, 0)
p = p.vLine(w/2.0)
p = p.threePointArc((L/2.0, w/2.0 + t), (L, w/2.0))
p = p.vLine(-w/2.0)
p = p.mirrorX()
p = p.extrude(20.0, True)

#make the neck
p = p.faces(">Z")
p = p.workplane()
p = p.circle(3.0)
p = p.extrude(2.0, True)

#make a shell
result = p.faces(">Z").shell(1)
show_object(result)

```

Blender

Blender ist eine sehr ausgereifte Software, die für die 3D-Animation entwickelt wurde.

Leider ist die Bedienung recht ungewohnt für jemanden wie mich, der recht viel mit Sketchup

gemacht hat.

Man kann nicht einfach mal so einen Polygonzug mit direkter eingabe von Maßen erstellen. Alles beginnt mit einem Quadrat und dieses erscheint an den „Cursor“-Koordinaten und liegt immer in der xy-Ebene. Dann kann man die Kanten und Punkte verschieben, Kanten/Flächen in regelmäßige Abschnitte unterteilen. Wie gesagt, recht gewöhnungsbedürftig. Bei einigen Operationen kann man per Tastatur direkt Maßangaben machen, was das exakte Zeichnen wiederum möglich macht.

Scheinbar ist es per Python beliebig erweiterbar, wodurch theoretisch eine SketchUp-ähnliche Bedienung machbar sein sollte.

Blender 2.80

Mit der lang erwarteten Version von Blender wurde das Userinterface in vielen Teilen verbessert und für neue Benutzer leichter zugänglich gemacht. Ich hab im [Blender for Artists Forum](#) einen Beitrag gestartet, der Blender SketchUp gegenüberstellt. Für Blender 2.81 werden noch bessere Snapping-Tools für präzises Arbeiten erwartet, siehe developer.blender.org.

Fork: Mechanical Blender

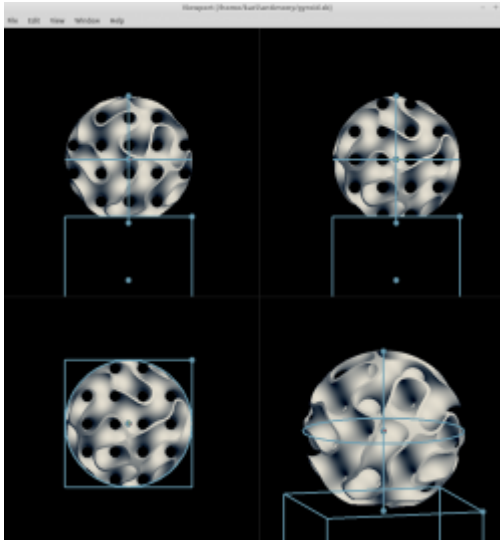
Ist eine sehr interessante Erweiterung, die Entwicklung ist aber noch nicht ausgereift und scheint auch eher still zu stehen. Mehr dazu hier: [Mechanical Blender](#).

Antimony

Matt Keeter ist ein ambitionierter Wissenschaftler und Programmierer, der sich mit 3D-Modellierung durch Funktionale auseinandersetzt. Eines seiner Werke ist [Antimony](#). Dabei zeichnet man einen Objektgraphen, der sehr intuitiv anpassbar ist. In einer 4-fach-Ansicht werden die 3 Projektionen und eine 3D-Perspektive live gerendert.

Die verwendbaren Blöcke sind schon recht umfangreich, und können durch eigene Skripts sehr einfach erweitert werden.

Hier zu sehen: eine Minimalfläche Namens Gyroid (Alan Schoen, 1970), verschnitten mit einer Kugel und einem Quader, sodass das Ding nicht davonrollt:



Sketchup

Für die Sachen, die ich mit einem 3D-CAD Programm machen möchte, reichte Sketchup bisher in den meisten Fällen aus, und es ist wirklich sehr schön zu bedienen. Ein Beispiel von mir ist die [Stiege im Selbstbau](#).

Derzeit läuft bei mir unter Linux Mint 18.1 Sketchup 2013 unter Wine recht brauchbar.

Auch die Make-Version im Browser läuft sehr gut. Die Funktionen wurden hier aber noch weiter beschnitten (z.B. der Outliner ist in der kostenlosen Version nicht verfügbar).

Externe Artikel zum Thema

- <https://www.sculpteo.com/blog/2016/07/05/top-7-best-3d-modeling-sofware-for-3d-printing-linux/>

CAD für Windows/MacOS

Einige sehr nette Programme gibt es für die populäreren Betriebssysteme:

- Autodesk Fusion 360 (ca. 700€/Jahr): sehr beliebt in der 3D-Druckerszene, weil kostenlos nutzbar für Hobby-Projekte
- Mol 3D (Kaufpreis 300\$): Bezier Modellierung, sehr einsteigerfreundlich, vom Macher von Rhino
- Rhino (Kaufpreis 1000€): parametrisiertes Modellieren für Künstler

[deutsch](#), [software](#), [cad](#), [video](#), [linux](#), [sammlung](#)

From:

<http://www.zeilhofer.co.at/wiki/> - **Verschiedenste Artikel von Karl Zeilhofer**

Permanent link:

http://www.zeilhofer.co.at/wiki/doku.php?id=3d_cad_on_linux

Last update: **2020/07/24 22:31**

